

**IMPLEMENTASI HIGH AVAILABILITY SERVER DENGAN  
TEKNIK FAILOVER VIRTUAL  
COMPUTER CLUSTER**

**MAKALAH**

PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA



Diajukan oleh:

*Irfani*

*Hernawan Sulistyanto, S.T.,M.T.*

**PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA  
JUNI 2015**

**HALAMAN PENGESAHAN**

Publikasi ilmiah dengan judul:  
**IMPLEMENTASI HIGH AVAILABILITY SERVER DENGAN  
TEKNIK FAILOVER VIRTUAL  
COMPUTER CLUSTER**

Yang dipersiapkan dan disusun oleh:

**Irfani**

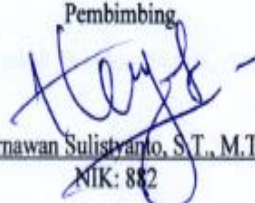
**L200110027**

Telah disetujui pada:

Hari : Kamis .....

Tanggal : 25 Juni 2015 .....

Pembimbing

  
Hernawan Sulistyanto, S.T., M.T.  
NIK: 842

Publikasi ilmiah ini telah diterima sebagai salah satu persyaratan


Untuk memperoleh gelar sarjana

Tanggal: 27 Juli 2015 .....

Mengetahui,

Ketua program studi

Informatika

  
Heru Supriyono, S.T., M.Sc., Ph.D.  
NIK: 970



UNIVERSITAS MUHAMMADIYAH SURAKARTA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448  
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: [informatika@fki.ums.ac.id](mailto:informatika@fki.ums.ac.id)

**SURAT KETERANGAN LULUS PLAGIASI**

**/A.3-II.3/INF-FKI/VII/2015**

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : IRFANI  
NIM : L200110027  
Judul : IMPLEMENTASI HIGH AVAILABILITY SERVER DENGAN  
TEKNIK FAILOVER VIRTUAL COMPUTER CLUSTER  
Program Studi : Informatika  
Status : **Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 7 Juli 2015

Biro Skripsi  
Informatika

**Adje Sapetra, S.Kom**



## Turnitin Originality Report

IMPLEMENTASI HIGH AVAILABILITY  
SERVER DENGAN TEKNIK FAILOVER  
VIRTUAL COMPUTER CLUSTER by Irfani

From publikasi september 2015 (publikasi)

Processed on 06-Jul-2015 10:50 WIB  
ID: 554227503  
Word Count: 2816

Similarity Index  
**17%**

## Similarity by Source

Internet Sources:	16%
Publications:	2%
Student Papers:	5%

## sources:

- 1 5% match (Internet from 05-Jul-2015)  
<http://pasca.unhas.ac.id/jurnal/files/a5c117f9997e173a7b9e52a28eb2d70.pdf>
- 2 2% match (Internet from 08-Jan-2013)  
<http://yayoi.com/2010/02/22/implementasi-konfigurasi-drbd-pada-suse-linux-enterprise-server-11-bagian-2/>
- 3 1% match (Internet from 11-Jun-2015)  
<http://sisilisilakasi.blogspot.com/>
- 4 1% match (student papers from 10-Mar-2015)  
Class: publikasi  
Assignment:  
Paper ID: 514603789
- 5 1% match (Internet from 03-Feb-2015)  
<http://www.seclab.ac.id/online/download/Server>
- 6 1% match (Internet from 18-Sep-2013)  
<http://eprints.unsri.ac.id/view/subjects/QA76.html>
- 7 1% match (student papers from 25-Feb-2012)  
Submitted to Universiti Putra Malaysia on 2012-02-25
- 8 1% match (Internet from 15-Mar-2014)  
<http://www.scribd.com/doc/192404443/Implementasi-High-Availability-Using-Teknologi-Virtual-Computer-Cluster>
- 9 1% match (Internet from 02-Feb-2015)  
<http://yopiwijaya88.blogspot.com/2014/04/sistem-informasi-penyalaran-bibit-untuk.html>
- 10 1% match (Internet from 20-Jan-2014)  
<http://mamedrehanjo.blogspot.com/>
- 11 < 1% match (Internet from 30-Jun-2015)  
[http://www.researchgate.net/publication/273135230\\_IMPLEMENTASI\\_FAILOVER\\_MENGGUNAKAN\\_JARINGAN\\_YFN\\_DAN\\_MITRON](http://www.researchgate.net/publication/273135230_IMPLEMENTASI_FAILOVER_MENGGUNAKAN_JARINGAN_YFN_DAN_MITRON)
- 12 < 1% match (student papers from 19-Jun-2014)  
Class: publikasi maret 2014  
Assignment:  
Paper ID: 435664491
- 13 < 1% match (Internet from 22-Mar-2010)  
[http://www.ijcsi.org/papers/IJCSI\\_6-2-25-32.pdf](http://www.ijcsi.org/papers/IJCSI_6-2-25-32.pdf)
- 14 < 1% match (Internet from 21-May-2015)  
<http://digilib.uin-suka.ac.id/3382/1/BAB%20V.pdf>
- 15 < 1% match (Internet from 11-Oct-2013)  
<http://digilib.uin-suka.ac.id/8826/>
- 16 < 1% match (Internet from 19-Jun-2015)  
[http://www.researchgate.net/publication/273135230\\_IMPLEMENTASI\\_FAILOVER\\_MENGGUNAKAN\\_JARINGAN\\_YFN\\_DAN\\_MITRON](http://www.researchgate.net/publication/273135230_IMPLEMENTASI_FAILOVER_MENGGUNAKAN_JARINGAN_YFN_DAN_MITRON)
- 17 < 1% match (Internet from 14-Oct-2013)  
<http://eprints.mdp.ac.id/298/1/Analisis%20Kandungan%20Informasi%20Pengumuman%20Stock%20Sulit%20Di%20Bursa%20Efek%20>

# **IMPLEMENTASI HIGH AVAILABILITY SERVER DENGAN TEKNIK FAILOVER VIRTUAL COMPUTER CLUSTER**

**Irfani**

Program Studi Informatika, Fakultas Komunikasi dan Informatika

Universitas Muhammadiyah Surakarta

Email: vnbuseng1@gmail.com

## **ABSTRAK**

Kebutuhan sistem informasi yang *up to date* mendorong penyedia layanan *web server* untuk membangun sistem dengan tingkat *availability* yang tinggi. Teknik yang bisa digunakan yaitu dengan menerapkan *server clustering*. *Cluster server* merupakan teknologi yang menggabungkan beberapa sumber daya yang bekerja bersama-sama sehingga tampak seolah-olah merupakan suatu sistem tunggal. Salah satu teknik *cluster* yang dapat digunakan untuk menyediakan sistem dengan *availability* yang tinggi yaitu *failover cluster*.

Penelitian ini bertujuan merancang sistem *failover virtual computer cluster* sebagai salah satu solusi untuk mengatasi kegagalan fungsi *server* dengan menggunakan VMware Workstation 11 sebagai platform simulasinya. *Failover virtual cluster* yang dibangun terdiri dari dua buah *server virtual* dengan sistem operasi Ubuntu Server 14.10. Kedua *server* ter-install aplikasi *heartbeat* yang berfungsi menghubungkan kedua *server* dan aplikasi DRBD yang berfungsi sebagai sinkronisasi data.

Parameter yang diukur pada penelitian ini adalah *Availability*, *Downtime*, CPU *Utilization*, dan *Throughput*. Berdasarkan hasil penelitian tersebut diperoleh nilai *availability* paling besar mencapai 99,50% dengan tingkat kestabilan *cluster* dari sisi CPU *utilization* dan *Throughput*, sehingga sistem *cluster virtual* ini dapat menjadi solusi untuk meningkatkan sistem dengan tingkat *availaibility* yang tinggi.

Kata kunci : *Availability, Cluster Server, Failover Cluster*

## PENDAHULUAN

Teknologi *cluster computer* merupakan salah satu teknologi yang memanfaatkan beberapa sumber daya komputer tunggal untuk bekerja bersama-sama sehingga tampak seperti satu sistem yang saling terintegrasi. Teknologi *cluster* memiliki kelebihan yaitu bisa menghasilkan suatu sistem dengan tingkat realibilitas tinggi dan sistem yang memiliki tingkat *availability* tinggi atau biasa disebut *high availability server*.

Teknologi ini digunakan untuk mengantisipasi kegagalan atau kerusakan *devices* pada komputer *server* yang dapat mengganggu kinerja sistem jaringan. Gangguan yang muncul pada sistem jaringan disebabkan karena *server* utama mati dan tidak ada *server backup* yang menggantikan fungsi *server* utama yang mati sehingga proses komunikasi antar jaringan terganggu. Salah satu solusi untuk mengatasi masalah tersebut dengan menggunakan teknologi *failover clustering server*.

Fungsi utama *failover clustering* (Hirt, 2009) bertujuan untuk membantu menjaga akses *client* ke aplikasi dan sumber daya *server*, bahkan ketika terjadi kegagalan *software*, ataupun kegagalan fungsi *server* yang mengakibatkan *server* berhenti bekerja.

Teknologi ini diharapkan dapat menjadi solusi dalam mengatasi kegagalan *server* ketika terjadi gangguan ataupun perawatan (*maintenance*).

Pada skripsi ini peneliti akan mengimplementasikan suatu sistem *failover virtual computer cluster* untuk mengatasi kegagalan *server*. Sistem yang dibangun merupakan *prototype* dengan memanfaatkan virtualisasi menggunakan VMware pada sistem operasi *windows* yang berfungsi sebagai *host* dan sistem operasi pada *guest server* menggunakan sistem operasi *Ubuntu 14.10*. Sistem ini kemudian akan dianalisa dengan melakukan beberapa pengujian untuk mengetahui tingkat *availability* dan performanya.

## TINJAUAN PUSTAKA

Febriani (2011) melakukan penelitian *failover virtual computer server* menggunakan *Windows Server 2008 R2 x64* sebagai OS-nya dan *Hyper-V* sebagai platfoirm simulasinya. *Test* performa dilakukan untuk lima parameter yang berbeda yaitu *processor arithmetic*, *.NET arithmetic*, *memory bandwidth*, *memory latency* dan *cache* dan memori. Hasil penelitian terhadap lima parameter tersebut menunjukkan hasil bahwa sistem *failover virtual computer cluster* tidak lebih baik

dibandingkan dengan sistem yang tidak di *cluster*. Sedangkan dari segi *availability* saat terjadi *failure* sistem *cluster* server lebih baik dibandingkan dengan sistem yang tidak di *cluster*.

Muchtar, dkk (2012) melakukan penelitian *failover cluster* pada dua *platform* yang berbeda dengan sistem operasi Linux dan Windows. Dari penelitiannya didapatkan hasil bahwa sistem *failover cluster* pada *Promox* menggunakan UCARP dan DNS *failover* dengan distro Linux sama baiknya dari segi kinerja dalam mencapai nilai standar *availability fault tolerance* yang telah ditetapkan yaitu 99,99%, namun DNS *failover* sedikit lebih baik karena dapat digunakan pada kedua *platform* Linux ataupun Windows.

Budi (2013) melakukan penelitian implementasi *virtual cluster* berbasis *cloud computing*. Penelitian dilakukan menggunakan *Hypervisor Promox Virtual Environment* (Promox VE) dengan platform virtualisasinya menggunakan *Kernel-Based Virtual Machine* (KVM). Hasil penelitiannya didapatkan bahwa sistem *high availability cluster* dapat meminimalisir kemungkinan *downtime* dengan meningkatkan *uptime server* yang cukup signifikan dibandingkan dengan *server* konvensional.

Patil, dkk (2014) melakukan penelitian *failover cluster* dengan menggabungkan database ber-*license* untuk sistem utama dan database *freeware* untuk sistem cadangan. Sistem *failover cluster* tersebut dibangun berdasarkan pertimbangan *Cost Effective Failover Clustering* (CEFC). Hasil penelitian tersebut didapatkan bahwa dengan menggunakan CEFC biaya pengadaan sistem *failover clustering* jauh lebih efektif dan efisien jika dibandingkan dengan sistem yang konvensional.

Lwin dan Thein (2009) melakukan penelitian berjudul “*High Availability System for Local Disaster Recovery with Markov Modeling Approach*”. Penelitian tersebut menganalisa perbedaan waktu *downtime* antara *Semi-Markov Process* (SMP) dan *Continuous Time Markov Chain* (CTMC). Analisa *availability* menggunakan *tool Sharpe* dengan hasil penelitian waktu *downtime* model CTMC sedikit lebih baik dibandingkan dengan model SMP.

## METODOLOGI

Pada penelitian ini, digunakan metode penelitian eksperimental. Menurut Sukardi (2011) penelitian eksperimen merupakan metode sistematis guna membangun hubungan



yang mengandung fenomena sebab akibat (*causal-effect relationship*). Sedangkan menurut Sugiyono (2012) penelitian eksperimen adalah penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi yang terkendalikan.

Pengaruh perlakuan yang dimaksud dalam penelitian ini adalah pengaruh penggunaan teknologi *failover virtual computer cluster* terhadap *availability server*. Implementasi dan uji coba dilakukan pada komputer berbasis sistem operasi linux dan *platform virtual* menggunakan VMware, kemudian dilakukan pengukuran *availability* dengan memberikan beberapa gangguan dan pengujian pada *server*. Hal ini dilakukan secara berulang-ulang, kemudian dilakukan pengukuran berdasarkan parameter yang diperoleh sebelumnya.

*Failover cluster* yang dibangun menggunakan dua buah *server virtual* yang berfungsi sebagai *server* utama dan *server* cadangan serta menggunakan tiga buah *network* untuk saling berkomunikasi didalam *cluster* tersebut.

Detail ketiga *network* yang akan dibuat sebagai berikut:

1. *Network* 1: 192.168.1.114 (*server* utama).
2. *Network* 2: 192.168.1.178 (IP *virtual heartbeat*).

3. *Network* 3: 192.168.1.119 (*server* cadangan).

Kedua *server virtual* yang dibangun memiliki spesifikasi *hardware* dan *software* sebagai berikut:

**Tabel 1.** Spesifikasi Komputer *Server*

Hardware	Tipe	Ukuran
Mainboard	Gigabyte	
Processor	AMD Athlon (tm) II X2 240	2.80
Memory	DDR3 PC 1600	2 GB
Hardisk	Serial ATA	250 GB
NIC	PCI-e Gigabit Ethernet	

Software yang digunakan untuk membuat virtualisasi *failover cluster* sebagai berikut:

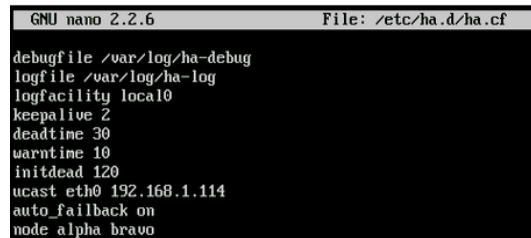
1. Sistem operasi: Windows 7 (*host*), Ubuntu Server 14.10 (*guest/server virtual*).
2. *Software* virtualisasi: VMware Workstation 11.
3. *Failover cluster*: Heartbeat
4. *Web server*: Apache web server.
5. *Backup* dan *recovery*: Distributed Replicated Block Device (DRBD).
6. *Software Benchmarking*: httpperf

Setelah semua data yang dibutuhkan terpenuhi maka proses instalasi dan konfigurasi *failover virtual cluster* siap dilakukan dengan melalui dua tahapan utama sebagai berikut :

1. Instalasi dan konfigurasi *software* pendukung.



- a. Instalasi sistem operasi host komputer. Sistem operasi yang digunakan untuk *host* komputer adalah Windows 7 x64.
  - b. Instalasi virtualisasi VMware Workstation 11. File instalasi bisa di-download melalui web resmi VMware dengan alamat <http://vmware.com/>.
  - c. Instalasi *guest server virtual* Ubuntu Server 14.10. File instalasi bisa di-download melalui web resmi ubuntu dengan alamat <http://ubuntu.com/download/server> /.
  - d. Instalasi *web server*. Web server yang digunakan untuk kedua *server* adalah *apache web server*. Instalasi dapat dilakukan melalui *repository* ubuntu dengan perintah **#apt-get install apache2**.
2. Instalasi dan konfigurasi *failover cluster*.
    - a. Instalasi dan konfigurasi *heartbeat*. Instalasi dapat dilakukan melalui *repository* ubuntu dengan perintah **#apt-get install heartbeat**. Setelah proses instalasi selesai, tahap selanjutnya adalah konfigurasi *heartbeat*.
    - b. Membuat *file* *ha.cf* pada kedua *server* dengan perintah **#nano /etc/ha.d/ha.cf**. Hasilnya seperti Gambar 1 di bawah ini.



```

GNU nano 2.2.6                                File: /etc/ha.d/ha.cf
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
warntime 10
initdead 120
ucast eth0 192.168.1.114
auto_failback on
node alpha bravo
  
```

**Gambar 1.** Konfigurasi *ha.cf*

- c. Konfigurasi *auth keys* dengan perintah **#nano /etc/ha.d/authkeys**. Isi konfigurasi dengan:

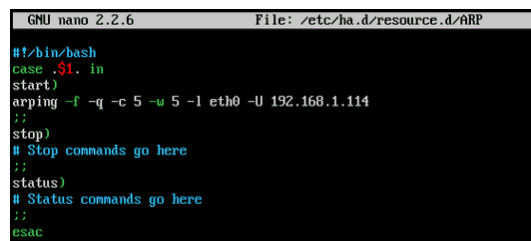
```
Auth 1
1 sha1 testing01
```

Ubah *file permission* dengan perintah **#chmod 600 /etc/ha.d/authkeys**

- d. Konfigurasi IP *virtual* dengan perintah **#nano /etc/ha.d/haresources**. Isi *file* konfigurasi dengan:

```
Alpha IPAddr::192.168.1.178/24/eth0
apache2
```

- e. Konfigurasi *Address Resolution Protocol* (ARP) dengan perintah **#nano /etc/ha.d/resource.d/ARP** kemudian atur *file permission* dengan perintah **#chmod +x /etc/ha.d/resource.d/ARP**



```

GNU nano 2.2.6                                File: /etc/ha.d/resource.d/ARP
#!/bin/bash
case "$1" in
start)
arping -f -q -c 5 -u 5 -I eth0 -U 192.168.1.114
;;
stop)
# Stop commands go here
;;
status)
# Status commands go here
;;
esac
  
```

**Gambar 2.** Konfigurasi ARP

f. Restart *heartbeat* kedua *server* *virtual* dengan perintah

**# /etc/init.d/heartbeat restart.**

### 3. Instalasi dan konfigurasi DRBD

Sebelum melakukan proses instalasi DRBD, perlu ditambahkan *harddisk* pada masing-masing *server* Alpha dan *server* Bravo sebesar 2Gb, *harddisk* ini akan digunakan sebagai *harddisk cluster* DRBD. Setelah ditambahkan cek dengan perintah **# fdisk -l**. Hasilnya seperti Gambar 3 di bawah ini.

```
root@alpha:/home/alpha# fdisk -l

Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xaae4def0

Device Boot Start End Sectors Size Id Type
/dev/sda1 * 2048 499711 497664 243M 83 Linux
/dev/sda2 501758 41940991 41439234 19.8G 5 Extended
/dev/sda5 501760 41940991 41439232 19.8G 8e Linux LVM

Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x348f3494

Device Boot Start End Sectors Size Id Type
/dev/sdb1 2048 4194303 4192256 2G 83 Linux

Disk /dev/mapper/ubuntu--vg-root: 19.2 GiB, 20661141504 bytes, 40353792 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk /dev/mapper/ubuntu--vg-swap_1: 508 MiB, 532676608 bytes, 1040384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@alpha:/home/alpha#
```

**Gambar 3.** Penambahan Hardisk

#### Instalasi DRBD

a. Instalasi paket DRBD dapat dilakukan melalui *repository* Ubuntu pada kedua *server* dengan perintah:

**#apt-get install drbd8-utils drbdlinks**

b. *Edit file* *drbd.conf* dengan perintah **#etc /etc/drbd.conf**. Isi *file* tersebut dengan:

```
1 | global {
2 |     dialog-refresh 1;
3 |     usage-count yes;
4 |     minor-count 5;
5 | }
6 | common {
7 |     sycer {
8 |         rate 10M;
9 |     }
10 | }
11 | resource r0 {
12 |     protocol C;
13 |     disk {
14 |         on-io-error detach;
15 |     }
16 |     sycer {
17 |         rate 10M;
18 |         al-extents 257;
19 |     }
20 |     on alpha {
21 |         device /dev/drbd0;
22 |         address 192.168.1.114:7788;
23 |         meta-disk internal;
24 |         disk /dev/sdb;
25 |     }
26 |     on bravo {
27 |         device /dev/drbd0;
28 |         address 192.168.1.119:7788;
29 |         meta-disk internal;
30 |         disk /dev/sdb;
31 |     }
32 | }
```

c. Buat *meta data disk* kedua *server* *virtual* dan jalankan *service* DRBD dengan perintah:

```
Drbdadm create-md r0
Service drbd start
```

d. Ubah *server* utama menjadi *primary server* dengan perintah:

```
#drbdsetup /dev/drbd0 primary --overwrite-
data-of-peer
```

Cek status DRBD sehingga hasilnya seperti Gambar 4 di bawah ini.

```
root@alpha:/home/alpha# drbdsetup /dev/drbd0 primary --overwrite-data-of-peer
root@alpha:/home/alpha# service drbd status
drbd driver loaded OK; device status:
version: 8.4.3 (api:1/proto:86-101)
srcversion: 107E17F432E25ED3AF8929
n:res cs ro ds p mounted fstype
... sync'ed: 13.1% (1825132/2097852)K
0:r0 SyncSource Primary/Secondary UpToDate/Inconsistent C
```

**Gambar 4.** Status DRBD

## HASIL DAN PEMBAHASAN

### 1. Analisa *Availability*

*Availability* analisis diukur berdasarkan istilah “*nine*”, dimana semakin banyak “*nine*” maka semakin tinggi sebuah sistem *availability*. Istilah “*nine*” digunakan karena tidak ada sistem yang mendapatkan ketersediaan seratus persen. *Availability* dapat digunakan sebagai salah satu parameter dalam *Service Level Agreement* (SLA). SLA adalah perjanjian yang disepakati antara penyedia layanan dengan pengguna dalam ruang lingkup untuk menentukan karakteristik dan kualitas layanan yang akan diberikan (ITU-T, 2002).

Braastad (2006) dan Calzolari (2009) seperti berikut.

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

- Mean Time Between Faults* (MTBF) merupakan waktu rata-rata *uptime*.
- Mean Time to Repair* (MTTR) merupakan rata-rata waktu yang diperlukan untuk mengembalikan layanan.

Hasil pengujian *availability* dapat dilihat pada Tabel 2 di berikut ini

**Tabel 2.** Hasil Pengujian *Availability*

NO	Waktu Downtime	Penyebab Kegagalan	MTBF (menit)	MTTR (menit)	Availability
1.	06 April 2015	Web Server Down	2235	0,5	99,97%
2.	10 April 2015	Web Server Down	4672	0,7	99,98%
3.	12 April 2015	Update Sistem Operasi	2744	1,4	99,94%
4.	17 April 2015	Web Server Down	7139	1,0	99,98%
5.	18 April 2015	Listrik Mati	1732	112	93,92%
6.	20 April 2015	Web Server Down	2810	0,6	99,97%
7.	23 April 2015	Listrik Mati	3660	16	99,56%
8.	24 April 2015	Web Server Down	1580	0,4	99,97%
Nilai Availability Total			26572	132	99,50%

Pengujian *availability* pada penelitian ini tidak mengambil waktu sebelum terjadi kegagalan sistem ataupun sesudah kegagalan sistem. Tetapi mengamati apakah layanan *server* tetap berjalan saat terjadi kegagalan pada salah satu *server virtual*.

Perhitungan *availability cluster* berdasarkan rumus yang digunakan oleh

Berdasarkan nilai *benchmark* pada Tabel 2 untuk analisa pengujian *availability* hasilnya mendekati 0 dan dapat dikatakan *zero downtime*. Karena *delay* yang terjadi pada pengujian *availability* tersebut memiliki nilai rata-rata *delay* dalam 20 hari sebesar 16,5 menit dengan tingkat *availability* total mencapai 99,50%.

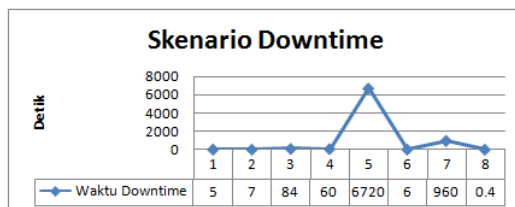
## 2. Analisa Parameter *Downtime*

*Downtime* adalah waktu (*period of time*) dimana sistem tidak dapat digunakan untuk menjalankan fungsinya sesuai yang diharapkan. *Downtime* sangat berpengaruh pada nilai *availability* dari suatu *equipment*, tetapi belum tentu berpengaruh pada *Loss Time*.

Hasil pengujian pada skenario *downtime* diambil dari pengujian *availability* dengan menghitung antara waktu sebelum *failover* atau *Mean Time Between Faults* (MTBF) sampai dengan waktu sebelum mengembalikan layanan atau *Mean Time to Repair* (MTTR). Hasil pengujiannya seperti Tabel 3 di bawah ini.

**Tabel 3.** Hasil Pengujian *Downtime*

NO	Penyebab Kegagalan	Failover (menit)	Failback (menit)
1.	Web Server Down	2235	2235,5
2.	Web Server Down	4672	4672,7
3.	Update Sistem Operasi	2744	2745,4
4.	Web Server Down	7139	7140
5.	Listrik Mati	1732	1844
6.	Web Server Down	2810	2810,6
7.	Listrik Mati	3660	3676
8.	Web Server Down	1580	1580,4



**Gambar 5.** Grafik Hasil Pengujian *Downtime*

Berdasarkan hasil pengujian layanan *web server* pada Tabel 3 dan

Gambar 5 menunjukkan bahwa besarnya persentase *availability server* mempengaruhi lamanya waktu *downtime* yang dialami *cluster server*. Semakin besar tingkat *availability server* maka semakin cepat yang dibutuhkan oleh *server cluster* untuk mengembalikan layanan atau *failback* ketika terjadi kegagalan pada *server* utama, begitu juga sebaliknya.

Waktu *failback* tercepat didapatkan ketika *server* mengalami kegagalan akibat *web server downtime* dengan waktu *failover* 1580 detik dan waktu yang dibutuhkan untuk mengembalikan layanan atau *failback* adalah 1580,4 detik dengan lama waktu *downtime* 4 detik. Waktu terlama *downtime* terjadi ketika *server* mengalami gangguan listrik mati yaitu dengan lama *downtime* 6720 detik.

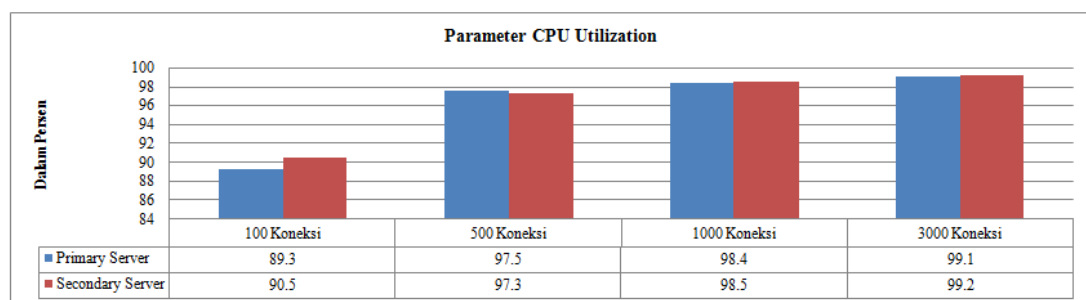
## 3. Analisa Parameter CPU Utilization

CPU *usage* atau CPU *time* dari sebuah program adalah lama waktu penggunaan prosesor yang diperlukan program untuk menjalankan instruksi-instruksinya. Pengujian analisa CPU *Utilization* dilakukan pada kedua sisi *server cluster* dengan menggunakan *httpperf*. Nilai CPU *Utilization* diperoleh dari beberapa kali pengujian *web server* dengan mengambil data pada presentase CPU *time*.

Hasil perbandingan skenario yang digunakan antara kedua server pengujian CPU *Utilization* pada kedua tidak jauh berbeda karena kedua server *server virtual* dapat dilihat pada Tabel 4 tersebut menggunakan *hardware* yang dan Gambar 7 di bawah ini. sama.

**Tabel 4.** Hasil Pengujian CPU *Utilization* Server Virtual

Skenario Pengujian		Primary Server					Secondary Server					Satuan
		Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	
Jumlah	100	88,4	89,3	88,6	79,5	87,6	80,3	85,6	71,7	90,0	90,5	Persen
	500	84,8	96,6	97,4	97,5	97,1	97,3	97,3	97,3	97,0	97,3	
	1000	98,4	98,4	98,4	98,4	98,4	98,5	98,3	98,4	98,3	98,0	
	3000	99,1	99,1	99,1	99,1	99,1	99,1	99,1	99,1	99,1	99,2	



**Gambar 6.** Perbandingan CPU *Utilization* Server Virtual

Hasil pengujian CPU *Utilization* pada Tabel 4 dan grafik pada Gambar 6 menunjukkan bahwa semakin berat beban pengujian maka mempengaruhi nilai CPU *Utilization* yang digunakan. Dari Tabel 4 dan grafik pada Gambar 6 nilai CPU *Utilization* paling kecil didapat dari skenario pengujian 100 koneksi pada *server backup* dengan CPU *usage* yang digunakan sebesar 71,7%. Sedangkan nilai CPU *Utilization* paling besar didapatkan dari skenario pengujian 5000 koneksi pada *server backup* dengan nilai 99,3%.

Dari hasil pengujian tersebut terlihat juga bahwa perbedaan nilai CPU

#### 4. Analisa Parameter *Throughput*

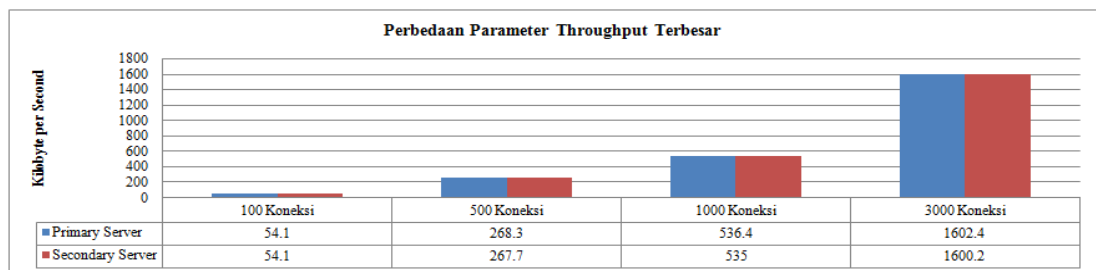
*Throughput* adalah nilai rata-rata pada pengiriman pesan yang sukses melalui sebuah *link* komunikasi. Data tersebut dapat dikirim melalui sebuah *link physical* maupun *logical*, atau lewat sebuah *network node* tertentu. Satuan untuk nilai *throughput* adalah *bit* per detik (*bit/s* atau *bps*).

Pada pengujian analisis *throughput* dilakukan beberapa kali percobaan dengan mengirimkan beban *request* yang berbeda dan semakin meningkat. Semakin tinggi nilai *throughput*, maka semakin bagus performa *server* tersebut. Hasil analisis

parameter *throughput* dapat dilihat pada Tabel 5 dan grafik pada Gambar 7 di bawah ini.

**Tabel 5.** Hasil Parameter *Throughput* Server Virtual

Analisa Throughput		Primary Server					Secondary Server					Satuan
	Jumlah Koneksi	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	
	100	54.0	54.1	54.0	54.1	53.9	53.8	54.1	53.4	53.9	54.0	Kilobyte per Second
	500	268.0	267.9	267.4	268.3	266.0	267.0	266.8	266.9	267.7	267.5	
	1000	534.6	534.2	534.2	532.5	536.4	534.6	531.8	535.0	532.8	534.4	
	3000	1602.4	1596.4	1602.1	1600.9	1594.4	1600.2	1593.8	1536.3	1598.1	1598.4	



**Gambar 7.** Perbandingan Parameter *Throughput* Server Virtual

Grafik pada Gambar 7 terlihat bahwa terjadi penurunan nilai *throughput* antara *server* utama dan *server* cadangan, nilai tersebut dipengaruhi oleh kecepatan inisiasi *heartbeat* saat memindahkan layanan dari *server* utama ke *server* cadangan. Dari grafik pada Gambar 8 tersebut juga diperoleh nilai *throughput* terkecil dari pengujian 1000 koneksi pada *server* cadangan dengan nilai *throughput* 522,2 Kb/s. Sedangkan nilai *throughput* terbesar diperoleh dari pengujian 100 koneksi pada *server* utama dengan nilai *throughput* sebesar 524,8 Kb/s.

## KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem *failover cluster* yang dibangun dapat bekerja dengan baik sesuai dengan konsep kerjanya, sehingga *client* tetap dapat mengakses layanan yang disediakan oleh *server* melalui *server* cadangan.
2. *Heartbeat* adalah aplikasi yang digunakan untuk *failover cluster* yang berfungsi untuk menentukan *server* utama dan *server* cadangan, serta menentukan IP *virtual*.

3. Hasil pengujian *availability cluster* sangat dipengaruhi oleh lama durasi on dari cluster tersebut. Nilai *availability* paling besar yaitu 99,92% dengan waktu delay 16 detik.
4. Hasil pengujian performa *failover cluster*, terlihat kualitas layanan antara *server* utama dan *server* cadangan tidak jauh berbeda karena proses sinkronisasi data menggunakan DRBD, sehingga dapat disimpulkan bahwa *server* cadangan dapat menjaga kualitas layanan yang disediakan oleh *cluster server*.
5. DRBD adalah aplikasi yang digunakan untuk proses sinkronisasi data antar *server* pada *level block device*. DRBD bekerja di bawah sistem dan dapat dikatakan tidak mengganggu sistem apabila sistem sedang berjalan.
6. Hasil pengujian parameter CPU *Utilization* dan *throughput* didapatkan hasil dengan perbedaan yang tidak terlalu signifikan antara *server* utama dan *server* cadangan. Hal tersebut dikarenakan kedua *server* menggunakan perangkat keras yang sama pada sistem virtualisasinya.
7. Konfigurasi *cluster* menunjukkan hasil yang baik dilihat dari segi *availability* pada saat terjadi *failure*.

## Saran

1. Berdasarkan penelitian yang telah dilakukan disarankan pada penelitian selanjutnya untuk menggunakan *hardware* yang lebih mendukung seperti komputer intel Xeon untuk menambah kemampuan kerja sistem.
2. Untuk instansi pendidikan, seperti sekolah dan universitas, sistem virtualisasi dapat dijadikan solusi untuk melakukan penelitian dengan berbagai sistem operasi tanpa harus menyediakan mesin *server* untuk sebuah sistem operasi.
3. Penggunaan *failover cluster* secara *virtual* dapat dikembangkan bagi penyedia layanan *server* untuk menyediakan sistem dengan ketersediaan yang tinggi serta dapat menghemat biaya penyediaan perangkat keras *server*.
4. Selain *Failover Cluster*, *Load Balancing web server* bisa menjadi solusi untuk ketersediaan server yang tinggi dengan terlebih dahulu dilakukan penelitian untuk hal tersebut.
5. Perlu pertimbangan lebih lanjut mengenai penggunaan virtualisasi dalam menyediakan *cluster virtual*. Hal tersebut berkaitan dengan penghematan biaya jika dibandingkan dengan *cluster* yang tidak di-*virtual*.



## DAFTAR PUSTAKA

- Budi, Setiya. 2013. *Implementasi Cluster Pada Web Server Berbasis Cloud Computing. Skripsi.* Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga. Yogyakarta.
- Calzolari, Federico; Arezzini, Silvia; Ciampa, Alberto eds. *High Availability Using Virtualization.* IOP Science, 2010.
- Don Frima, Iyoga.(2012). *Implementasi Sistem Multiple-Computer Cluster Menggunakan Linux Enterprise Real Application Cluster (LINUXERAC) berbasis Metode Storage Area Network (DRBD) serta Analisa High Performance dan High Availability.* Depok : Universitas Indonesia.
- Febriani, Tania Rizky. 2011. *Implementasi dan Analisa Sistem Failover Virtual Computer Server. Skripsi.* Fakultas Teknik Universitas Indonesia.
- Hidayat, Fikri. 2012. *Implementasi dan Analisa Redudansi dan High Availability Dalam Server Untuk Diskless Thin Client Berbasis Storage Area Network. Skripsi.* Fakultas Teknik Universitas Indonesia.
- Hirt, Alan. 2009. *Pro SQL Server 2008 Failover Clustering,* Apress, New York.
- ITU-T Standardization E.860. 2002. *Framework of a Service Level Agreement.*
- Lukitasari, Desi & Oklilas, Ahmad Fali. 2010. *Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server.* Universitas Sriwijaya : Jurnal Generic Fakultas Ilmu Komputer.

Lwin dan Thein. (2009). *High Availaibility Cluster System for Local Disaster Recovery with Markov Modeling Approach. International Journal of Computer Science Issues*. Vol 6, No 2.

Muchtar dkk. *Implementasi Failover Clustering Pada Dua Platform Yang Berbeda Untuk Mengatasi Kegagalan Fungsi Server*. Fakultas Teknik Universitas Hasanuddin.

Patil, N.V., et al. (2014). *Cost Effective Failover Clustering. International Journal of Research in Engineering and Technology*. Vol 3: hal 3.

Peraturan Menteri Komunikasi Dan Pemerintah, 2007 ,Pasal 27 Standar Kualitas Layanan Dan Kinerja Jaringan Untuk Layaanan Berbasis Internet Protokol.

Pratama, I Putu Agus Eka. 2014. *Smart City Beserta Cloud Computing dan Teknologi-teknologi Pendukung Lainnya*. Bandung:Informatika.

Purbo, W.O.2011. *Kluster Komputer*. ([http://opensource.telkomspeedy.com/wiki/index.php/ Kluster\\_komputer](http://opensource.telkomspeedy.com/wiki/index.php/Kluster_komputer), diakses 5 April 2015).

Rasian, R dan Mursanto, P. 2009. *Perbandingan Kinerja Pendekatan Virtualisasi*. Jurnal Sistem Informasi. Magister Teknik Informatika Univeristas Indonesia.

Software Development for High Availability Cluster : [www.drbd.org](http://www.drbd.org). Oleh LINBIT HA- Solutions. Vienna, Austria. 2008-2011. diakses pada Bulan Februari 2012.

- Sugianto, Masin Vavai. 2011. *“Kelebihan dan Kekurangan Virtualisasi.*  
(<https://www.excellent.co.id/product-services/vmware/keuntungan-teknologi-virtualisasi-cloud-computing>, diakses 5 April 2015).
- Sugiyono. 2012. *Statistik Untuk Penelitian*. Bandung : Alfabeta.
- Sujatmiko, Eko. 2012. *Kamus Teknologi Informasi dan Komunikasi.*  
Surakarta:Aksarra Sinergi Media.
- Sundarranjan, S., S. Bhattacharya. 2006. *Xen and Server Consolidation*, Infosys  
Whitepaper.
- Zamzami, Nurul Fadilah. 2012. *Implementasi Load Balancing dan Failover Menggunakan Mikrotik Router OS Berdasarkan Multihomed Gateway Pada Warung Internet "DIGA"*. Skripsi. Program Studi Telekomunikasi Politeknik Telkom.